Departamento de Engenharia de Eletrónica e Telecomunicações e de Computadores
Licenciatura em Engenharia Informática e de Computadores

Second Practical Project

Artificial Intelligence course

2022/2023 Summer Semester

Version 1.00

Teacher: Nuno Leite

ISEL, 25 April 2023

ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Artificial Intelligence course
Second Practical Project
2022/2023 Summer Semester

Learning objectives

At the end of the **Second practical project**, students should be able to:

- ☐ Approach Problem-Solving as Search, using basic (non-informed) and more advanced algorithms (informed);
- ☐ Understand the concept of *state space*;
- ☐ Apply non-informed and informed search algorithms to solve the Sudoku game;
- ☐ Approach Problem-Solving as search in a solution space;
- ☐ Understand optimisation algorithms that explore a space of solutions, such as *simulated annealing* (SA) and *genetic algorithms* (GA);
- ☐ Apply *simulated annealing* and *genetic algorithms* to solve the Sudoku game.

Sudoku game

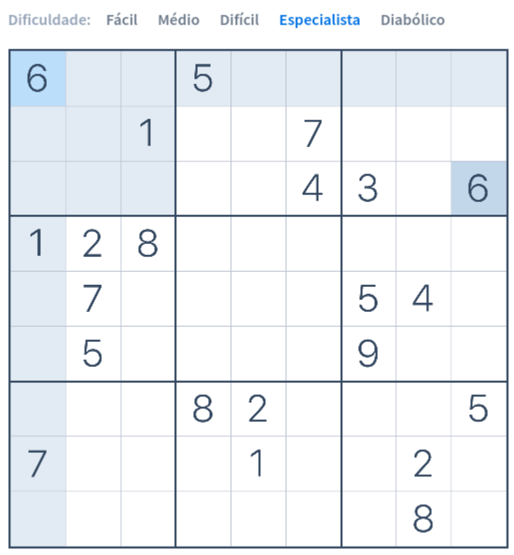
In this project, your group will program an *automated solver* for the famous Sudoku game. The word Sudoku literally means "digit-single". The following is a description based from the Wikipedia[1]: Sudoku is a logic-based, combinatorial number-placement puzzle. The objective of the classic game set is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 subgrids that compose the grid (also called "boxes", "blocks", or "regions") contain all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a single solution.

Figure 1 illustrates an example of a Sudoku puzzle (*expert* level) from `https://sudoku.com/pt/expert/`.

Your assignment is to develop programs (in Prolog and in other language of your choice) that demonstrate the behaviour of different solvers, namely solvers that employ the following algorithms:

- iterative-deepening (non-informed), programmed in Prolog;
- best-first search or A* (informed), programmed in Prolog. You have to devise a proper *admissible heuristic*;
- simulated annealing (optimisation algorithm), programmed in a language of your choice;

[1] `https://en.wikipedia.org/wiki/Sudoku`



6			5					
		1			7			
					4	3		6
1	2	8						
	7					5	4	
	5					9		
			8	2				5
7				1			2	
							8	

Figure 1: Example of a Sudoku puzzle (*expert* level).

- genetic algorithms (optimisation algorithm), programmed in a language of your choice.

These algorithms will be available in the GitHub's Artificial Intelligence course site, and are written in Prolog and Matlab languages.

The input/output is text-based.

The representation of the *state/solution/chromosome* (respectively, in state-space search/SA/GA) should be devised carefully, leading to simple and efficient operators.

You should demonstrate the behaviour of the automated solvers for some example levels, ranging from simple puzzles to very difficult puzzles (all the puzzles should be well posed). You don't need to implement level loaders (the levels could be hard-coded for simplicity), but if you want to do so, you could use the level format published in: `https://github.com/t-dillon/tdoku/blob/master/data.zip`.

Due date: 23 May 2023 until 23:59.

The delivery of the work must present the report, Prolog code (solver using search algorithms), and other developed code (solver using optimization algorithms, *simulated annealing* and *genetic algorithms*), delivered in the Moodle system. The report must be concise, explain the main predicates/structure, and justify all decisions taken. It must indicate the student group composition and the curricular unit info.

The report must present experimental results on several types of puzzles, and a comparison of the execution time of the solvers should be done.